AB-36

APPLICATION
BRIEF

80186/80188 DMA Latency

STEVE FARRER
APPLICATIONS ENGINEER

April 1989

Order Number: 270525-001

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

80186/80188 DMA LATENCY

CONTENTS

PAGE

When using the DMA controller of the 80186 and 80188, there are several operating conditions which affect the service time (latency) between when the DMA request is generated and when the bus cycles associated to the DMA transfer are actually run. This application brief describes those conditions which affect DMA Latency.

DMA REQUEST GENERATION

The minimum DMA latency is 4 clocks and, depending on when the signal arrives (i.e. if the signal just missed the setup time), it might appear to be almost 5 clocks. This 4 to 5 clock delay is due to a two phase synchronizer and various transfer gate delays the DRQ signal must take before reaching the BIU. Conceptually the circuit looks like Figure 1.

If the Bus Interface Unit (BIU) is available when the DRQ signal reaches it, then a DMA cycle will proceed at T1 of the bus cycle as the next clock.

Also note that the DRQ signal is not latched, and must remain active until serviced. If the DRQ signal is brought low after being asserted high, then a '0' will propagate through and; if the request had not yet been serviced, then the BIU will see a '0' and the cycle will never take place.

Conditions Affecting DMA Latency

The circumstances that affect DMA latency in order of worst case are as follows:

1) HOLD
2) LOCK - INTA
3) Odd byte accesses
4) Effective Address Calculations (EA)

HOLD can indefinitely delay a DMA cycle. There is no mechanism internally to remove HLDA when a DMA request is pending.

LOCKed instructions can also delay a DMA cycle by a significant amount, depending on the type of instruction locked. A typical locked XCHG instruction from memory to register could delay the DMA cycle by as much as 18 clocks if the memory access required two bus cycles (80188 or odd locations on the 80186). On the other hand, a locked repeat MOVS could delay a DMA cycle by up to 1.05 million clocks depending on the number of transfers and the number of bus cycles per transfers.

Interrupt acknowledges can also affect DMA latency because the bus is locked out during the first two bus cycles required to fetch the interrupt vector type. This causes the worst case latency during interrupt acknowledges to be:

4	Clocks (Minimum Setup)
10	Clocks (2 Bus Cycles + 2 Idle Clocks) Min
14	Clocks Total

Both HOLD and LOCK are extremely dependent on the type of system being designed and therefore are not really considered to be normal worst case latency. However, odd byte accesses and effective address calculations are conditions that frequently occur in almost all systems. Under these conditions of no HOLD, no LOCK, and no wait states, the worst case occurs when the DMA request loses to an instruction data cycle requiring an effective address calculation.

Effective addresses (EA) always require 4 clocks for calculation and can only take place during T3-T4-TI-TI, T4-TI-TI-TI, or TI-TI-TI-TI. This creates an extra minimum insertion of 2 T-idle cycles. If the EA requires an immediate value in the prefetch queue, then a signal goes active which places the EA bus cycle at a higher priority than any other BIU requests. This is so the execution unit won't be waiting on the bus interface unit. If the EA hadn't required the value in the queue, then the EU could proceed with the next instruction shortly after it had sent the request to the BIU. Figure 2 shows the effects EA calculations have on DMA Latency.

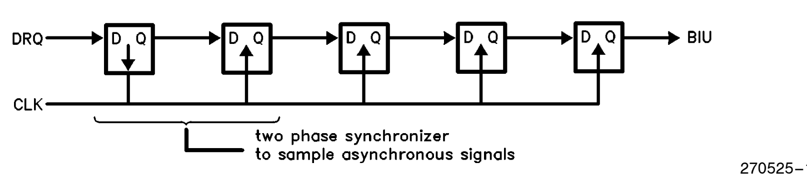

Figure 1. DMA Request Synchronization

Address	Code	Instruction
FA058	90	NOP
FA059	90	NOP
FA05A	2E87060100	XCHG AX, CS:WORD PTR 0001

Fetch of last byte necessary for the XCHG instruction. (269–272)

18 CLOCKS (269–286)

	A/D	STATUS				
269	FA05E	1100	ALE	T1	11	recognized on edge
270	0A05E	0100	FETCH	T2	11	DRQ active
271	0A000	0111	PASSIVE	T3	11	
272	0A000	0111	PASSIVE	T4	11	
273	FA05F	1100	ALE	T1	11	DRQ presented to BIU (too late to stop fetch)
274	0A05F	0100	FETCH	T2	11	
275	0A0BE	0111	PASSIVE	T3	11	EA calculation (275–278)
276	0A0BE	0111	PASSIVE	T4	11	
277	0A0BE	0111	PASSIVE	T3	11	*
278	0A0BE	0111	PASSIVE	T4	11	
279	FA011	1101	ALE	T1	11	Fetch or memory read cycle for the XCHG instruction. (279–286)
280	0A011	0101	READ	T2	11	
281	0A001	0111	PASSIVE	T3	11	
282	0A001	0111	PASSIVE	T4	11	
283	FA012	1101	ALE	T1	11	
284	0A012	0101	READ	T2	11	
285	0A001	0111	PASSIVE	T3	11	
286	0A001	0111	PASSIVE	T4	11	
287	F800C	1101	ALE		11	DMA cycle begins
288	1800C	0101	READ		11	
289	180FF	0111	PASSIVE		11	
290	180FF	0111	PASSIVE		11	

270525–2

* If an immediate value had not been used, the EA would have been aborted and the DMA would have begun its bus cycle. In this case, the latency would be 8 clocks.

Figure 2. Logic State Analyzer Trace and Accompanying Program Code

INTEL CORPORATION, 2200 Mission College Blvd., Santa Clara, CA 95052; Tel. (408) 765-8080

INTEL CORPORATION (U.K.) Ltd., Swindon, United Kingdom; Tel. (0793) 696 000

INTEL JAPAN k.k., Ibaraki-ken; Tel. 029747-8511

Printed in U.S.A./xxxx/0296/B10M/xx xx